APSIPA Transactions on Signal and Information Processing, 2022, 11, e35

Original Paper

PAGER: Progressive Attribute-Guided Extendable Robust Image Generation

Zohreh Azizi* and C.-C. Jay Kuo

University of Southern California, Los Angeles, CA, USA

ABSTRACT

This work presents a generative modeling approach based on successive subspace learning. Unlike most generative models in the literature, our method does not utilize neural networks to analyze the underlying source distribution and synthesize images. The resulting method, called the progressive attribute-guided extendable robust image generative (PAGER) model, has advantages in mathematical transparency, progressive content generation, lower training time, robust performance with fewer training samples, and extendibility to conditional image generation. PAGER consists of three modules: core generator, resolution enhancer, and quality booster. The core generator learns the distribution of low-resolution images and performs unconditional image generation. The resolution enhancer increases image resolution via conditional generation. Finally, the quality booster adds finer details to generated images. Extensive experiments on MNIST, Fashion-MNIST, and CelebA datasets are conducted to demonstrate generative performance of PAGER.

Keywords: Image generation, image synthesis, progressive generation, attribute-guided generation, successive subspace learning.

*Corresponding author: Zohreh Azizi, zazizi@usc.edu.

Received 31 May 2022; Revised 23 August 2022
ISSN 2048-7703; DOI 10.1561/116.0000034

1 Introduction

Unconditional image generation has been a hot research topic in the last decade. In image generation, a generative model is trained to learn the image data distribution from a finite set of training images. Once trained, the generative model can synthesize images by sampling from the underlying distribution.

GANs have been widely used for unconditional image generation with impressive visual quality in recent years [18]. Despite the evident advantages of GANs, their training is a non-trivial task: GANs are sensitive to training hyperparameters and generally suffer from convergence issues [23]. Moreover, training GANs requires large-scale GPU clusters and an extensive number of training data [46]. Limited training data usually cause the discriminator to overfit and the training to diverge [30]. These concerns have led to the development of improved GAN training methods [19], techniques for stabilized training with fewer data [30, 46], or non-adversarial approaches [23]. Yet, the great majority of existing generation techniques utilize deep learning (DL), a method for learning deep neural networks, as the modeling backbone.

A neural network is typically trained using a large corpus of data over long episodes of iterative updates. Therefore, training a neural network is often a time-consuming and data-hungry process. To ensure the convergence of deep neural networks (DNNs), one has to carefully select (or design) the neural network architecture, the optimization objective (or the loss) function, and the training hyper-parameters. Some DL-based generative models like GANs are often specifically engineered to perform a certain task. They cannot be easily generalized to different related generative applications. For example, the architectures of these neural networks for unconditional image generation have to be re-designed for image super-resolution or attribute-guided image generation. Last but not the least, due to the non-linearity of neural networks, understanding and explaining their performance is a standing challenge.

To address the above-mentioned concerns, this paper presents an alternative approach for unconditional image generation based on successive subspace learning (SSL) [37–40]. The resulting method, called progressive attribute-guided extendable robust image generative (PAGER) model, has several advantages, including mathematical transparency, progressive content generation, lower training time, robust performance with fewer training samples, and extendibility to conditional image generation.

PAGER consists of three modules: (1) core generator, (2) resolution enhancer, and (3) quality booster. The core generator learns the distribution of low-resolution images and performs unconditional image generation. The resolution enhancer increases image resolution via conditional generation. Finally, the quality booster adds finer details to generated images.

To demonstrate the generative performance of PAGER, we conduct extensive experiments on MNIST, Fashion-MNIST, and CelebA datasets. We show

that PAGER can be trained in a fraction of the time required for training DL based models and still achieve a similar generation quality. We then demonstrate the robustness of PAGER to the training size by reducing the number of training samples. Next, we show that PAGER can be used in image super resolution, high-resolution image generation, and attribute-guided face image generation. In particular, the modular design of PAGER allows us to use the conditional generation modules for image super resolution and high-resolution image generation. The robustness of PAGER to the number of training samples enables us to train multiple sub-models with smaller subsets of data. As a result, PAGER can be easily used for attribute-guided image generation.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. The PAGER method is proposed in Section 3. Experimental results are reported in Section 4. Extendability and applications of PAGER are presented in Section 5. Finally, concluding remarks and possible future extensions are given in Section 6.

2 Related Work

2.1 DL-based Image Generative Models

DL-based image generative models can be categorized into two main classes: adversarial-based and non-adversarial-based models. GANs [18] are adversarial-based generative models that consist of a generator and a discriminator. The training procedure of a GAN is a min-max optimization where the generator learns to generate realistic samples that are not distinguishable from those in the original dataset and the discriminator learns to distinguish between real and fake samples. Once the GAN model is trained, the generator model can be used to draw samples from the learned distribution. StyleGANs have been introduced in recent years. They exploit the style information, leading to better disentangability and interpolation properties in the latent space and enabling better control of the synthesis [31–33].

Examples of non-adversarial DL-based generative models include variational auto-encoders (VAEs) [34], flow-based models [14, 15], GLANN [23], and diffusion-based models [13, 21]. VAEs have an encoder/decoder structure that learns variational approximation to the density function. Then, they generate images from samples of the Gaussian distribution learnt through the variational approximation. An improved group of VAEs called Vector-Quantized VAEs (VQ-VAE) can generate outputs of higher quality. In VQ-VAEs, the encoder network outputs discrete codes and the prior is learnt instead of being static [54, 62]. Flow-based methods apply a series of invertible transformations on data to transform the Gaussian distribution into a complex distribution.

Following the invertible transformations, one can generate images from the Gaussian distribution. GLANN [23], employs GLO [4], and IMLE [45] to map images to the feature and the noise spaces, respectively. The noise space is then used for sampling and image generation. Recently, diffusion-based models are developed for image generation. During the training process, they add noise to images in multiple iterations to ensure that the data follows the Gaussian distribution ultimately. For image generation, they draw samples from the Gaussian distribution and denoise the data in multiple gradual steps until clean images show up.

Despite impressive results of DL-based generative models, they are mathematically not transparent due to their highly non-linear functionality. Furthermore, they are often susceptible to unexpected convergence problems [23], long training time, and dependency on large training dataset size. As we show in our experiments, PAGER addresses the aforementioned concerns while maintaining the quality of the images generated by DL-based techniques.

2.2 Unconditional and Conditional Image Generation

In unconditional image generation, sample images are drawn from an underlying distribution without any prior assumption on the images to be generated. In conditional image generation, samples are generated under a certain assumption. One example of the latter is the generation of a high-resolution image given a low-resolution image. The proposed PAGER method contains both unconditional and conditional image generation techniques. Its core generator module employs the unconditional image generation technique. Its resolution enhancer and quality booster modules perform conditional image generation. Although PAGER is an unconditional image generator by itself, it can be easily extended to conditional image generation with rich applications. We will elaborate this point with three examples, namely, attribute-guided face image generation, image super resolution, and high-resolution image generation. Each task is elaborated below.

2.2.1 Attribute-Guided Face Image Generation

For a set of required facial attributes, the goal is to generate face images that meet the requirements. Lu et al. [48] performs attribute-guided face image generation using a low-resolution input image. It modifies the original CycleGAN [69] architecture and its loss functions to take conditional constraints during training and inference. In Kowalski et al. [36], synthetic labeled data are used to factorize the latent space into sections which associate with separate aspects of face images. It designs a VAE with an additional attribute vector to specify the target part in the factorized latent space. Qian et al. [53] proposes to learn a geometry-guided disentangled latent space using facial landmarks

to preserve generation fidelity. It utilizes a conditional VAE to sample from a combination of distributions. Each of them corresponds to a certain attribute.

2.2.2 Image Super-resolution

The problem aims at generating a high-resolution image that is consistent with a low-resolution image input. One solution is the example-based method [17]. Others include auto-regressive models and normalized flows [52, 61, 63]. Quite a few recent papers adopt the DL methodology [16]. Another line of work treats super-resolution as a conditional generation problem, and utilize GANs or diffusion-based models as conditional generative tools which use low-resolution images as the generation condition [12, 41, 60].

2.2.3 Progressive Generation of Very-high-resolution Images

Generation of a very-high-resolution image of high quality is challenging and treated as a separate research track. A common solution is to take a progressive approach in training and generation to maintain the model stability and generation quality. There exist both GAN-based and diffusion-based very-high-resolution image generation solutions [21, 29].

Our PAGER method can be trained for unconditional image generation as well as for conditional image generation such as attribute-guided face image generation and image super-resolution. In principle, it can also be used for progressive generation of very-high-resolution images. Our PAGER serves as a general framework that can bridge different generation models and applications.

2.3 Successive Subspace Learning

In order to extract abstract information from visual data, spectral or spatial transforms can be applied to images. For example, the Fourier transform is used to capture the global spectral information of an image while the wavelet transform can be exploited to extract the joint spatial/spectral information. Two new transforms, namely, the Saak transform [39] and the Saab transform [40], were recently introduced by Kuo et al. [37–40] to capture joint spatial/spectral features. These transforms are derived based on the statistics of the input without supervision. Furthermore, they can be cascaded to find a sequence of joint spatial-spectral representations in multiple scales, leading to SSL. The first implementation of SSL is the PixelHop system [10], where multiple stages of Saab transforms are cascaded to extract features from images. Its second implementation is PixelHop++, where channel-wise Saab transforms are utilized to achieve a reduced model size while maintaining an effective representation [11]. An interesting characteristic of the Saab transform that makes SSL a good candidate for generative applications is that it is invertible. In other

words, the SSL features obtained by multi-stage Saab transforms can be used to reconstruct the original image via the inverse SSL, which is formed by multi-stage inverse Saab transforms. Once we learn the Saab transform from training data, applying the inverse Saab transform in inference would be trivial.[1]

SSL has been successfully applied to many image processing and computer vision applications [56]. Several examples include unconditional image generation [42–44], point cloud analysis [26–28, 65–68], fake image detection [7–9, 70], face recognition [57, 58], medical diagnosis [47, 51], low light enhancement [2], anomaly detection [64], to name a few. Inspired by the success of SSL, we adopt this methodology in the design of a new image generative model as elaborated in the next section.

2.4 SSL-based Image Generative Models

GenHop [42] is the contemporary SSL-based image generative model in literature. GenHop utilizes SSL for feature extraction. It applies independent component analysis (ICA) and clustering to obtain clusters of independent feature components at the last stage of SSL. Then, it finds a mapping between the distribution of ICA features and Guassian distributions. In this work, we do not perform ICA but model the distribution of SSL features via GMMs directly. As compared to GenHop, our approach offers several attractive features. First, it has lower computational complexity and demands less memory. Second, our method offers a progressive and modular image generation solution. It is capable of conditional and attribute-guided image generation. It can also be easily extended to other generative applications such as super-resolution or high-resolution image generation.

3 Proposed PAGER Method

The PAGER method is presented in this section. First, our research motivation is given in Section 3.1. Then, an overview on PAGER and its three modules are described in Section 3.2. Finally, our attribute-guided face image generation is elaborated in Section 3.3.

3.1 Motivation

A generative model learns the distribution of the training data in the training phase. During the generation phase, samples are drawn from the distribution as new data. To improve the accuracy of generative image modeling, gray-scale or color images should be first converted into dimension-reduced latent representations. After converting all training images into their (low-dimensional) latent

[1] https://github.com/zohrehazizi/torch_SSL

representation, the distribution of the latent space can be approximated by a multivariate Gaussian distribution. For learning the latent representation, most prior work adopts GAN-, VAE-, and diffusion-based generative models; they train neural networks that can extract latent representations from an image source through a series of nonlinear transformations. Similarly, we need to learn such a transformation from the image space to the latent representation space.

In this work, we utilize an SSL pipleline, rather than neural networks, to achieve the transformation to the latent representation space. The SSL pipeline consists of consecutive Saab transforms. In essence, it receives an image, denoted by $I \in \mathbb{R}^{w \times h \times c}$, and converts it into a latent feature vector, denoted by $X \in \mathbb{R}^n$, where w, h and c are the pixel numbers of the width, height and color channels of an image while n is the dimension of the latent vector. For the remainder of this paper, we refer to the latent space obtained by SSL as the *core space*. The Saab transform utilizes mean calculation and PCA computation to extract features from its input. Due to the properties of PCA, the i-th and j-th components in the core space are uncorrelated for $i \neq j$. This property facilitates the use of Gaussian priors for generative model learning over the core space.

Figure 1 illustrates the distributions of input image pixels (I) and Saab outputs (X). In this example, we plot the distributions of the first, second and third components of I (i.e., the RGB values of the upper-left pixel of all source images) and X (i.e., the Saab transform coefficients). The RGB components are almost uniformly distributed in the marginal probability. They are highly correlated as shown in the plot of joint distributions. In contrast, Saab coefficients are close to the Gaussian distribution and they are nearly uncorrelated. While the distributions of one- and two-dimensional components of X are very close to Gaussians, the distribution of higher-dimensional vectors might not be well modeled by one multivariate Gaussian distribution. For this reason, we employ a mixture of Gaussians to represent the distribution of the core space.

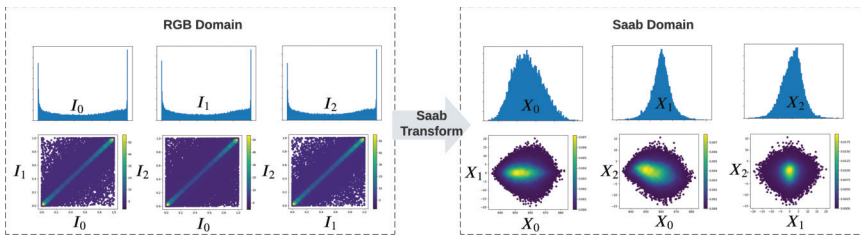

Figure 1: Example distributions from RGB pixels (left block) and Saab transforms (right block). The top figures correspond to single vector dimensions ($I_0 \ldots I_2$ in RGB and $X_0 \ldots X_2$ in Saab domains). The bottom figures correspond to joint distributions. Distributions are extracted from the first three components of CelebA images.

3.2 System Overview

An Overview of the PAGER generation method is shown in Figure 2. PAGER is an unconditional generative model with a progressive approach in image generation. It starts with unconditional generation in a low-resolution regime, which is performed by the core generator. Then, it sequentially increases the image resolution and quality through a cascade of two conditional generation modules: the resolution enhancer and the quality booster.

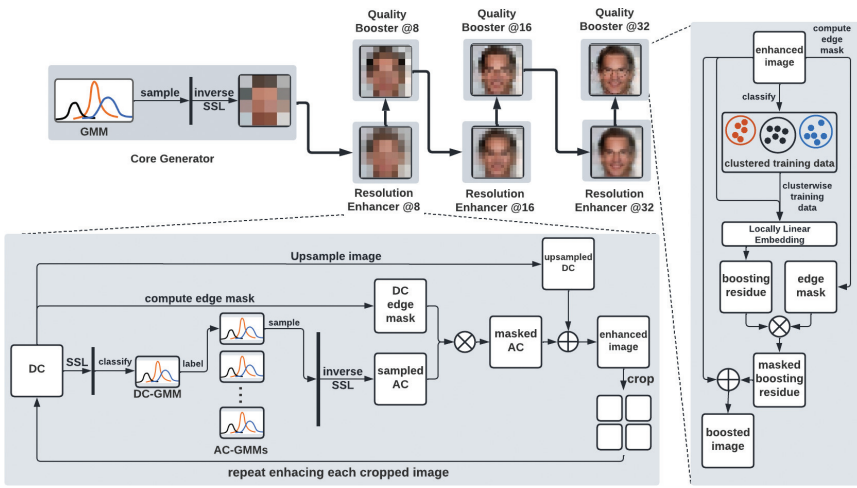

Figure 2: Overview of PAGER generation method.

3.2.1 Module 1: Core Generator

The core generator is the unconditional generative module in PAGER. Its goal is to generate low-resolution (e.g., $4 \times 4 \times 3$) color images. This module is trained with images of shape $2^d \times 2^d \times 3$ (e.g., $d = 2$). It applies consecutive Saab transforms on input images $\{I_i\}_{i=1}^M$ using PixelHop++ structure [11], ultimately converting images into n-dimensional vectors $X \in \mathbb{R}^n$ ($n = 2^d \times 2^d \times 3$) in core space. The goal of the core generator is to learn the distribution of $\{X_i\}_{i=1}^M$. We use $\mathcal{X}$ to denote a random variable within $\{X_i\}_{i=1}^M$, representing observed samples in core space. Let $P(\mathcal{X})$ be the underlying distribution of $\mathcal{X} \in \mathbb{R}^n$. The generation core G attempts to approximate the distribution $P(\mathcal{X})$ with a distribution $G(\mathcal{X})$.

DL-based methods utilize iterative end-to-end optimization of neural networks to achieve this objective. In PAGER, we model the underlying distribution of the core space using the Gaussian Mixture Model (GMM), which is highly efficient in terms of training time. This is feasible since we use

SSL to decouple random variables, which we illustrated in Section 3.1. The conjunction of multi-stage Saab (SSL) features and GMMs can yield a highly accurate density modeling. Formally, the GMM approximation of $G(\mathcal{X})$ is defined as follows:

$$G(\mathcal{X}) = \sum_{k=1}^{K} p_k \mathcal{N}(\mathcal{X}, \mu_k, \Sigma_k), \tag{1}$$

where $\mathcal{N}(\mathcal{X}, \mu_k, \Sigma_k)$ is a multi-variate normal distribution with mean μ_k and diagonal covariance matrix Σ_k, and p_k is a binary random variable. We have $p_k = 1$ with probability P_k, $p_k = 0$ with probability $(1 - P_k)$ and $\sum_{k=1}^{K} P_k = 1$. In other words, only one of the K Gaussian models will be selected at a time, and the probability of selecting the k-th Gaussian model is P_k in such a GMM. The parameters of the GMM can be determined using the Expectation Maximization (EM) algorithm [55]. Once such a GMM model is obtained, one can draw a sample, X, randomly and proceed to Modules 2 and 3.

The need for Modules 2 and 3 is explained below. $G(\mathcal{X})$ is learned from observations X_i, $i = 1 \cdots M$. When the dimension, n, of the core space is large, estimating $G(\mathcal{X})$ becomes intractable and the approximation accuracy of GMM would drop. For this reason, the unconditional generation process is constrained to a low-dimensional space. Then, we employ conditional generative models (modules 2 and 3) to further increase image resolution and quality.

3.2.2 Module 2: Resolution Enhancer

We represent image I_d as the summation of its DC and AC components:

$$I_d = DC_d + AC_d, \tag{2}$$

$$DC_d = U(I_{d-1}), \tag{3}$$

where I_d is an image of size $2^d \times 2^d$, U is the Lanczos image interpolation operator, DC_d is the interpolated image of size $2^d \times 2^d$ and AC_d is the residual image of size $2^d \times 2^d$. The above decoupling of DC and AC components of an image allows to define the objective of the resolution enhancer. It aims to generate the residual image AC_d conditioned on DC_d. In Figure 2, a multi-stage cascade of resolution enhancers is shown. The detail of a representative resolution enhancer is highlighted in the lower subfigure.

To train the resolution enhancer, we first decouple the DC and AC of training samples. Then, we extract SSL features from the DC and build a GMM model with K components, denoted by G_{DC}. By this method, we learn a distribution of the DC at a certain image resolution. Note that each DC from a training image belongs to one of the Gaussian models in G_{DC}. Therefore, DCs (and their associated AC) are clustered into K classes using G_{DC}. We gather the AC of each class and build a corresponding GMM,

denoted by $G_{AC,k}$ where $k \in \{1, \cdots, K\}$. In total, we learn $K+1$ GMMs: $\{G_{DC}, G_{AC,1} \ldots G_{AC,K}\}$.

At the test time, the resolution enhancer receives the low resolution image I_{d-1}, and upsamples it to obtain the interpolated DC, i.e., $DC_d = U(I_{d-1})$. Then, the resolution enhancer converts the DC to its SSL features and classifies it into one of the K clusters using G_{DC}. Mathematically, we have

$$X_{DC} = \text{SSL}(DC_d), \tag{4}$$

$$y = \arg_k \max \{\mathcal{N}(X_{DC}, \mu_k, \Sigma_k)\}_{k=1}^{K}, \tag{5}$$

where $\mathcal{N}(X_{DC}, \mu_k, \Sigma_k)$ is the probability score of X_{DC} according to the k-th component of G_{DC}, and the classification label y is the maximizer index. In other words, the resolution enhancer identifies a cluster of samples that are most similar to DC_d. Next, the resolution enhancer draws a sample from the AC distribution corresponding to class y:

$$X_{AC} \sim G_{AC,y}(\mathcal{X}_{AC}). \tag{6}$$

With the above two-step generation, the resolution enhancer generates X_{AC} conditioned on X_{DC}. Afterwards, X_{AC} is converted to the RGB domain using the inverse SSL transform:

$$AC_d = \text{SSL}^{-1}(X_{AC}). \tag{7}$$

The computed AC component is masked and added to the DC to yield the higher resolution image via

$$I_d = DC_d + \widehat{AC}_d, \tag{8}$$

$$\widehat{AC}_d = M(DC_d) \odot AC_d, \tag{9}$$

where $M(DC_d)$ is a mask and $\odot$ denotes element-wise multiplication. The mask is derived from the edge information obtained by the Canny edge detector [5]. The masking operation serves two objectives. First, it prevents details from being added to smooth regions of the DC component. Second, it suppresses unwanted noise. Once I_d is generated, it is cropped into four non-overlapping regions, and each region goes through another resolution enhancement process. The process is recursively applied to each sub-region to further enhance image quality. In our experiments, we continue the recursion until a cropped window size of 2×2 is reached.

3.2.3 Module 3: Quality Booster

The right subfigure of Figure 2 presents the quality booster module. It follows the resolution enhancer by adding detail and texture to the output of the

resolution enhancer. It exploits the locally linear embedding (LLE) [59] scheme and adds extra residue values that are missed by the resolution enhancer. LLE is a well known method in building correspondence between two components in image super resolution [6, 25] or image restoration [24]. To design the quality booster, we decompose the training dataset, enhance the DC component, and compute the residuals as follows:

$$I_d = DC_d + AC_d, \tag{10}$$

$$E_d = \text{Enhancer}(DC_d), \tag{11}$$

$$R_d = I_d - E_d, \tag{12}$$

where I_d represents a $2^d \times 2^d$ training image, E_d is the result of applying the enhancer module to the DC component of the image, and R_d is the residual image. During training, the quality booster stores E_d^i and R_d^i, $i = 1, \ldots, M$ from M training samples. In generation, the quality booster receives image E_d and uses the LLE algorithm to estimate the residual image for image E_d based on E_d^i and R_d^i from the training dataset. It approximates the residual image with a summation of several elements within R_d^i. Readers are referred to Roweis and Saul [59] for details of LLE computation. Similar to the enhancer module, the computed R_d^i is masked and added to E_d to boost its quality.

Although the LLE in the quality booster module uses training data residues during inference, it does not affect the generation diversity for two reasons. First, the quality booster only adds some residual textures to the image. In other words, it has a sharpening effect on edges. Since its role is limited to adding residuals and sharpening, it does not have a significant role in adding or preventing diversity. Second, the weight prediction mechanism of LLE provides a method to combine various patch instances and obtain diverse patterns.

3.3 Attribute-Guided Face Image Generation

In attribute-guided face image generation, the goal is to synthesize face images that have certain properties. Let $A \in \{-1, +1\}^T$ denote a set of T binary attributes. The goal is to synthesize an image that satisfies a query $\mathbf{q} \in \{-1, 0, +1\}^T$, where -1, 0, $+1$ denote negative, don't care, and positive attributes. For instance, if the attribute set is {*male*, *smiling*}, the query $\mathbf{q} = [-1, +1]$ requests an image of a female smiling person, and the query $\mathbf{q} = [0, -1]$ request an image (of any gender) that is not smiling.

Without loss of generality, we explain the attribute-guided generation process with $T = 7$. The attributes selected from attribute labels in CelebA dataset include "gender", "smiling", "blond hair", "black hair", "wearing lipstick", "bangs" and "young." Given these seven binary attributes, there are $2^7 = 128$ subsets of data that correspond to each unique set of selected attributes. However, some of the attribute combinations might not be abundant in the

training data due to the existing correlation between the attributes. For instance, "wearing lipstick", "bangs", and "gender" are highly correlated. Thus, instead of considering all 128 combinations, we partition the attributes of training data into K subsets using k-means clustering (we set $K = 10$ in our experiments). Based on the attribute clusters, we create K data subsets and train a separate PAGER model for each subset.

At generation time, the goal is to synthesize a sample with a given attribute set, $\mathbf{q} \in \{-1, 0, +1\}^7$. To determine which of the 10 models best represents the requested attribute set, we compute the Cosine distance of $\mathbf{q}$ to each of the cluster centers and select the model that gives the minimum distance. Then, we draw samples from the corresponding model. Figure 3 shows generated images corresponding to 15 different attribute vectors. We see that the attribute-based generation technique can successfully synthesize images with target attributes while preserving diversity and fidelity.

Figure 3: Examples of attribute-guided generated images for CelebA with various attribute combinations.

4 Experiments

4.1 Experimental Setup

We perform experiments on three datasets: MNIST, Fashion-MNIST, and CelebA. They are commonly used for learning unconditional image generative models. We briefly explain the experimental settings of PAGER for each dataset below.

4.1.1 CelebA

The dataset is a set of colored human face images. Suppose that there are $2^d \times 2^d$ pixels per image. To derive Saab features and their distributions, we apply d-stage cascaded Saab transforms. At each stage, the Saab filter has a spatial dimension of 2×2 with stride 2. The number of GMM components in the core generator is 500. The core generator synthesizes color images of size 4×4. Higher resolution images are generated conditioned on the previous resolution with the resolution enhancer and the quality booster modules in cascade ($4 \times 4 \rightarrow 8 \times 8 \rightarrow 16 \times 16 \rightarrow 32 \times 32$). The resolution enhancer has 100 GMM components for the DC part and 3 GMM components for the AC part at each stage. LLE in the quality booster module is performed using 2 nearest neighbors.

4.1.2 MNIST and Fashion-MNIST

The two datasets contain gray-scale images of digits and clothing items, respectively. The generation pipeline for these datasets is similar to CelebA except that the core generator synthesizes 16×16 padded gray-scale images for each of the 10 classes. The 16×16 images are converted to 32×32 with a single stage of resolution enhancer and quality booster. Finally, they are cropped to 28×28.

4.2 Evaluation of Generated Image Quality

4.2.1 Subjective Evaluation

We show image samples of resolution 32×32 generated by PAGER for MNIST, Fashion-MNIST and CelebA in Figure 4. Generated images learned from MNIST represent the structure of digits accurately and with rich diversity. Images generated from Fashion-MNIST show diverse examples for all classes with fine details and textures. Generated images for CelebA are semantically meaningful and with fine and diverse details in skin tone, eyes, hair and lip color, gender, hairstyle, smiling, lighting, and angle of view.

Figure 5 compares generated images by GenHop [42], which is an earlier SSL-based method, and our PAGER for the CelebA dataset. To be compatible with GenHop, we perform comparison on generated images of resolution 32×32. As seen, images generated by PAGER are more realistic with finer details than GenHop.

Next, we compare images generated by our method and those obtained by prior DL-based generative models in Figure 6. We resort our comparison to GAN [18], WGAN [1], LSGAN [50], WGAN-GP [19], GLANN [23], and diffusion-based model [22] of resolution 64×64. Note that these methods along with the selected resolution are ones that we could find over the Internet so as to

Figure 4: Examples of PAGER generated images for MNIST (top), Fashion-MNIST (middle), and CelebA (bottom) datasets.

allow a fair comparison to the best available implementations. Specifically, we take generated images of GAN, WGAN and LSGAN from celeba-gan-pytorch github.[2] We take those of WGAN-GP from WGAN-GP-DRAGAN-Celeba-Pytorch github.[3] For the diffusion model, we take the pre-trained model from pytorch-diffusion-model-celebahq github,[4] which generates samples of resolution 256×256. We resize generated samples to the resolution of 64×64 to make them comparable with other methods. Figure 6 compares generated images by prior DL-based generative models and our PAGER for the CelebA dataset. It can be seen that generated images of PAGER are comparable with

[2] https://github.com/joeylitalien/celeba-gan-pytorch
[3] https://github.com/joeylitalien/celeba-gan-pytorch
[4] https://github.com/FengNiMa/pytorch_diffusion_model_celebahq

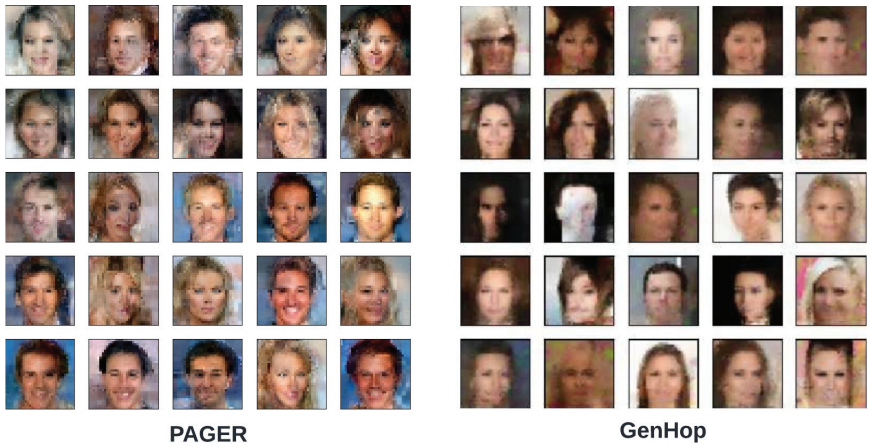

Figure 5: Example images generated by PAGER and GenHOP for the CelebA dataset.

those of prior DL-based methods. There are some noise patterns in our results. Their suppression is an interesting future research topic.

4.2.2 Objective Evalution

We use the Frechet Inception Distance (FID) [20] score to perform quantitative comparison of our method with prior art. FID is a commonly used metric to evaluate the performance of generative models. It considers both diversity and fidelity of generated images. We follow the procedure described in Lucic et al. [49] to obtain the FID scores; an Inception neural network extracts features from a set of 10K generated images as well as another set of 10K real (test) images. Two multivariate Gaussians are fit to the extracted features from two sets separately. Then, the Frechet distance between their mean vectors and covariance matrices is calculated. A smaller FID score is more desirable as it indicates a better match between the synthesized and real test samples.

The FID scores of various methods for MNIST, Fashion-MNIST and CelebA datasets are compared in Table 1. Methods in the first and second sections are both based on DL. Methods in the first section are adversarial generative models while those in the second section are non-adversarial. The results of the first and second sections are taken from [49] and [23], respectively. For the Diffusion model, we generated 10K samples using the pre-trained model from pytorch-diffusion-model-celebahq github[5] and measured the FID score. GenHop in Section 3 does not use a neural network backbone. Its results are taken from [42]. We see from Table 1 that the FID scores of PAGER are

[5] https://github.com/FengNiMa/pytorch_diffusion_model_celebahq

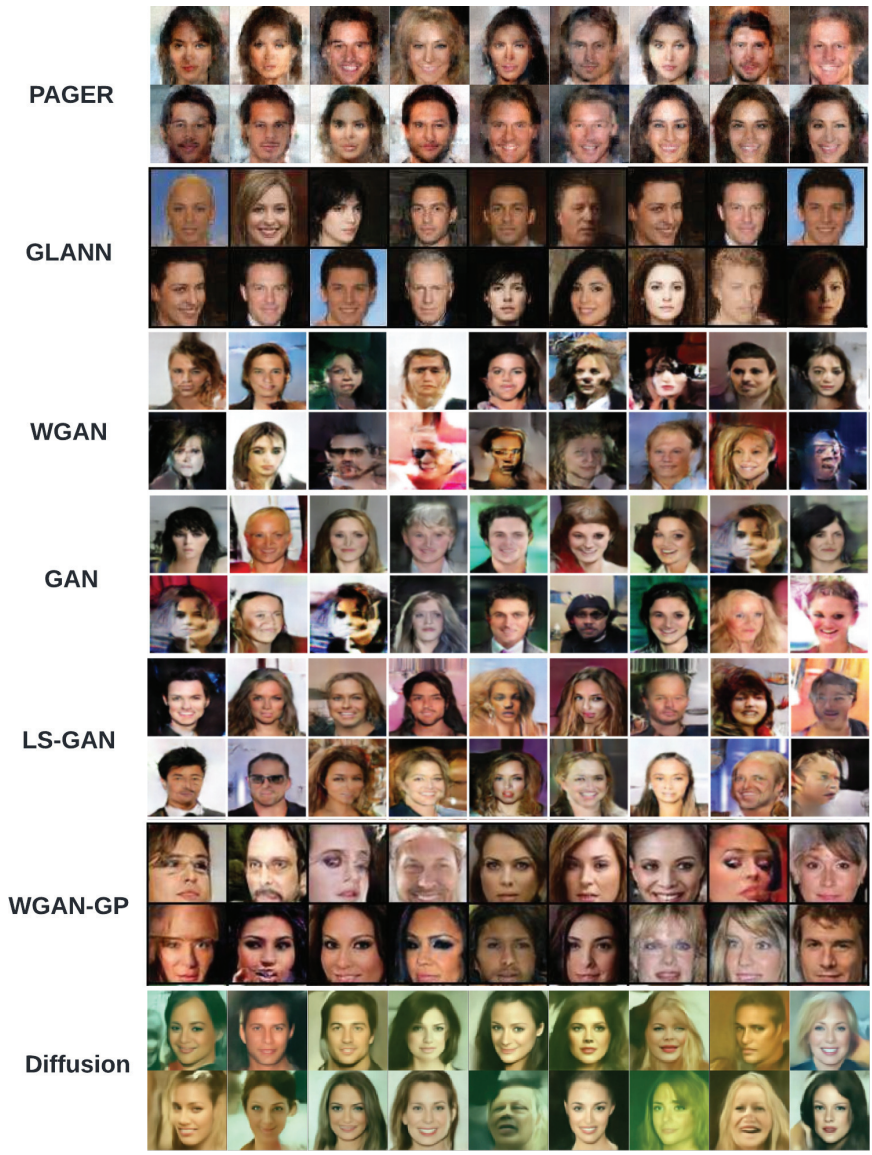

Figure 6: Samples generated by PAGER and prior DL-based generative models for the CelebA dataset.

Table 1: Comparison of FID scores for MNIST, Fashion-MNIST and CelebA datasets.

Method	MNIST	Fashion	CelebA
MM GAN [18]	9.8	29.6	65.6
NS GAN [18]	6.8	26.5	55.0
LSGAN [50]	7.8	30.7	53.9
WGAN [1]	6.7	21.5	41.3
WGAN-GP [19]	20.3	24.5	30.0
DRAGAN [35]	7.6	27.7	42.3
BEGAN [3]	13.1	22.9	38.9
VAE [34]	23.8	58.7	85.7
GLO [4]	49.6	57.7	52.4
GLANN [23]	8.6	13.0	46.3
Diffusion [22]	N/A	N/A	48.8
GenHop [42]	5.1	18.1	40.3
PAGER (Ours)	9.5	19.3	43.8

comparable with those of prior generative models. In training PAGER model for Table 1, we used 100K training images from CelebA and 60K training images from MNIST and Fashion-MNIST with no augmentation.

PAGER is still in its preliminary development stage. Although it does not outperform prior generative models in the FID score, it does have comparable performance in all three datasets, indicating its potential to be further improved in the future. In addition, PAGER has several other advantages to be discussed in the next subsection.

4.3 Other Performance Metrics

In this section, we study additional performance metrics: robustness to the number of training samples and training time.

4.3.1 Robustness to training dataset sizes

Figure 7 presents the FID score of PAGER and five DL-based generative models (MM GAN, LSGAN, WGAN, WGAN-GP, and GLANN) when the number of training samples is set to 1K, 2K, 5K, 10K, 20K, and 60K for MNIST dataset. To produce the FID scores of the GAN-based related work, we use the open-source implementation by PyTorch-GAN github.[6] For GLANN, we use the implementation provided by the authors. Since GLANN is not trained

[6] https://github.com/eriklindernoren/PyTorch-GAN

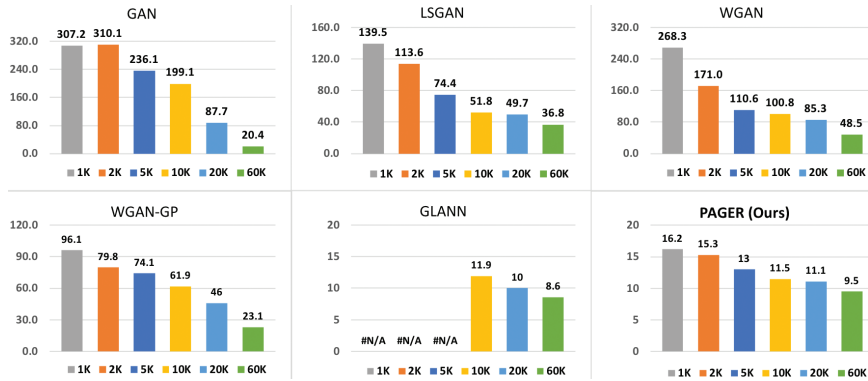

Figure 7: Comparison of FID scores of six benchmarking methods with six training sizes (1K, 2K, 5K, 10K, 20K, and 60K) for the MNIST dataset. The FID scores of PAGER are significantly less sensitive with respect to smaller training sizes.

with less than 10K samples, its FID scores for 1K, 2K and 5K samples are not available. It is worth noting that the FID scores for 60K training samples of some prior work in Figure 7 are different than those in Table 1. This happens because some of prior generative models (e.g., MM GAN, LSGAN, and WGAN) are too sensitive to training hyper-parameters and/or data augmentation [49]. The scores reported in Figure 7 are the best FID scores obtained using the default hyper-parameters in the open-source library. We see from Figure 7 that PAGER is least affected by the number of training samples. Even with the number of training samples as small as 1K, PAGER has an FID score of 16.2 which is still better than some prior works' original FID scores presented in Table 1, such as WGAN-GP, VAE and GLO. Among prior works, GLANN is less sensitive to training size but cannot be trained with less than 10K samples.

4.3.2 Comparison on Training Time

The training time of PAGER is compared with prior work in Table 2 on two platforms.

- CPU (Intel Xeon 6130): The CPU training time of PAGER is slightly more than 4 minutes, which is significantly less than all other methods as shown in Table 2. The normalized CPU training times of various DL-based methods against PAGER are visualized in the left subfigure of Figure 8. PAGER is 11× faster than WGAN and 325× faster than LSGAN.
- GPU (NVIDIA Tesla V100): The GPU training time of PAGER is around 3 minutes, which is again less than all other methods as shown

in Table 2. The normalized GPU run times of various methods are also visualized in the right subfigure of Figure 8. PAGER is 9× faster than WGAN and 48× faster than GLANN.

Table 2: Training time comparison.

Method	CPU	GPU
MM GAN [18]	93m14s	33m17s
LSGAN [50]	1426m23s	45m52s
WGAN [1]	48m11s	25m55s
WGAN-GP [19]	97m9s	34m7s
GLO [4]	1090m7s	139m18s
GLANN [23]	1096m24s	142m19s
GenHop [42]	6m12s	N/A
PAGER (Ours)	4m23s	2m59s

4.3.3 Joint Consideration of FID Scores and Training Time

To provide a better picture of the tradeoff between training time and FID score, we present both of these metrics in Figure 9. On this figure, points that are closer to the bottom left are more desirable. As seen, PAGER significantly outperforms prior art when considering FID scores and training time jointly.

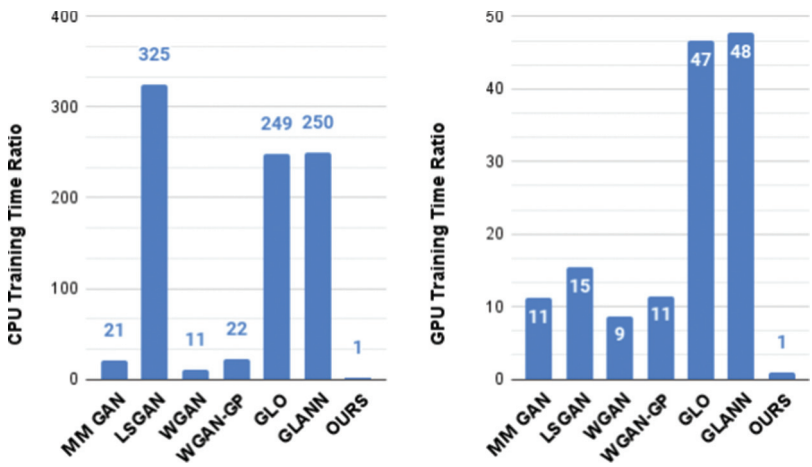

Figure 8: Comparison of normalized training time, where each bar represents the training time of a DL-based model corresponding to those shown in Table 2 and normalized by training time of PAGER.

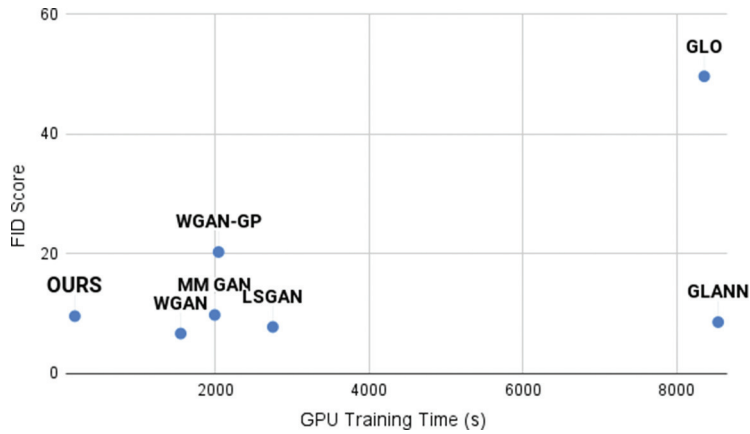

Figure 9: Comparison of joint FID scores and GPU training time of PAGER with DL-based related work in the generation of MNIST-like images. PAGER provides the best overall performance since it is closest to the left-bottom corner.

4.4 Discussion

Based on the above experimental results, we can draw the following conclusions.

- **Quality image generation.** The FID scores of PAGER are comparable with those of prior DL-based image generation techniques on common datasets. This indicates that PAGER can generate images of similar quality to prior art.
- **Efficient training.** PAGER can be trained in a fraction of the time required by DL-based techniques. For example, our MNIST generative model is trained in 4 minutes on a personal computer's CPU while the fastest prior work demands 25-minute training time on an industrial GPU. The efficiency of PAGER is achieved by the development of a non-iterative training scheme. CPU-based efficient training implies smaller energy consumption and carbon footprint than GPU-based DL methods. This is a major advantage of PAGER.
- **Robustness to training sample size.** PAGER can still yield images of reasonable quality even when the number of training samples is drastically reduced. For example, in Figure 10 we show that the number of training samples can be reduced from 100K to 5K with only a negligible drop in the generated image quality for the CelebA dataset.
- **Improvements over prior SSL-based generative model - GenHop.** While PAGER is the second SSL-based generative model, it is worthwhile to review its improvements over the prior SSL-based generative model

known as GenHop [42]. First, the great majority of CelebA generated samples by GenHop suffer from over-smoothing which blurs details and even fades out the facial components in many samples as shown in Figure 5. This is because GenHop heavily relies on LLE which has a smoothing effect and limits synthesis diversity. On the other hand, PAGER generates diverse samples with visible facial components. Note that PAGER only uses LLE to add residuals to already generated samples. It serves as a sharpening technique and does not affect synthesis diversity. Second, GenHop limits the resolution of generated samples to 32×32. This prevents GenHop to be extendable to high-resolution image generation or other generative applications like super-resolution. Third, GenHop takes longer time that PAGER to train and it is not implemented for GPU training. Fourth, GenHop only conducts unconditional image generation while PAGER has further applications such as attribute-guided image generation and super-resolution.

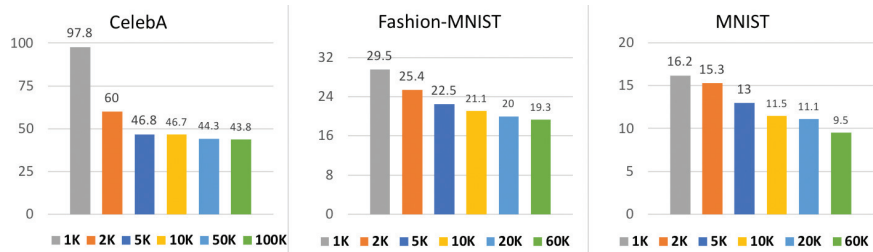

Figure 10: Comparison of PAGER's FID scores with six training sample sizes for CelebA, Fashion-MNIST and MNIST datasets. We see that the FID scores do not increase significantly as the training samples number is as low as 5K for CelebA and 1K for MNIST and Fashion-MNIST.

5 Comments on Extendability

In this section, we comment on another advantage of PAGER. That is, PAGER can be easily tailored to other contexts without re-training. We elaborate on three applications at the conceptual level.

- **Super Resolution.** PAGER's two conditional image generation modules (i.e., the resolution enhancer and the quality booster) can be directly used for image super resolution with no additional training. These modules enhance the image resolution from an arbitrary dimension $2^d \times 2^d$ to $2^{d+k} \times 2^{d+k}$, where k is the number of consecutive resolution enhancer and quality booster modules needed to achieve this task. Figure 11

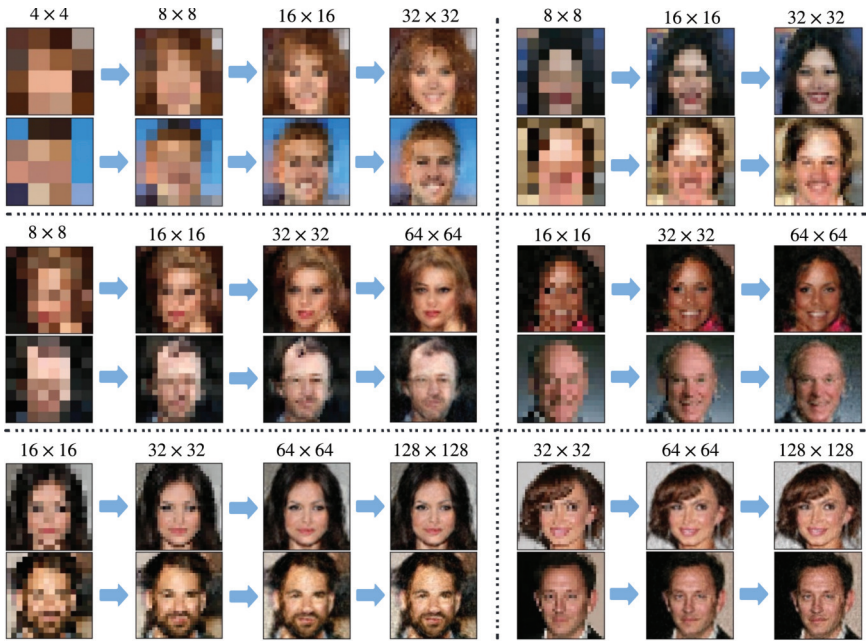

Figure 11: Illustration of PAGER's application in image super-resolution for CelebA images: Two top rows starting from resolution 4×4 (left block) and 8×8 (right block) and ending at resolution 32×32. Two middle rows starting from resolution 8×8 (left block) and 16×16 (right block) and ending at resolution 64×64. Two bottom rows starting from resolution 16×16 (left block) and 32×32 (right block) and ending at resolution 128×128.

shows several examples starting from different resolutions and ending at resolutions 32×32, 64×64 and 128×128.

- **Attribute-guided Face Image Generation.** To generate human face images with certain characteristics (e.g., a certain gender, hair color, etc.) we partition the training data based on the underlying attributes and construct subsets of data (Section 3.3). Each subset is used to train a different core generator that represents the underlying attributes. Examples of such attribute-guided face generation are presented in Figure 3. The feasibility of training PAGER using a subset of training data is a direct result of its robustness to the training dataset size. It was empirically evaluated in Figure 10. The mean FID score of CelebA-like image generation changes only 6% when the number of training samples is reduced from 100K to as low as 5K.

- **High-Resolution Image Generation.** PAGER can be easily extended to generate images of higher resolution. To achieve this objective, we can have more resolution enhancer and quality booster units in cascade to

(a) Resolution 128×128

(b) Resolution 256×256

Figure 12: Examples of generated CelebA-like images of resolution 128×128 and 256×256.

reach the desired resolution. We present several generated CelebA-like samples of resolution 128×128 and 256×256 in Figure 12. This gives some evidence that the current design of PAGER is extendable to higher resolution generation. On the other hand, to generate results comparable with state-of-the-art generative models like ProGAN [29], StyleGAN [31–33], VQ-VAE-2 [54] or diffusion-based models [13, 21], we need to further optimize our method. Further improvement on PAGER could lead to enhanced quality of generated images in higher resolutions.

6 Conclusion and Future Work

A non-DL-based generative model for visual data generation called PAGER was proposed in this work. PAGER adopts the successive subspace learning framework to extract multi-scale features and learns unconditional and conditional probability density functions of extracted features for image generation. The unconditional probability model is used in the core generator module to generate low-resolution images to control the model complexity. Two conditional image generation modules, the resolution enhancer and the quality booster, are used to enhance the resolution and quality of generated images

progressively. PAGER is mathematically transparent due to its modular design. We showed that PAGER can be trained in a fraction of the time required by DL-based models. We also demonstrated PAGER's generation quality as the number of training samples decreases. We then showed the extendibility of PAGER to image super resolution, attribute-guided face image generation, and high resolution image generation.

The model size of PAGER is primarily determined by the sizes of the quality booster. The number of parameters is about 46 million. The large quality booster size is due to the use of LLE in predicting residual details. We do not optimize the LLE component in the current implementation. As a future topic, we would like to replace it with a lightweight counterpart for model size reduction. For example, We might replace LLE with GMMs to learn the distribution of residual textures, to reduce the model size significantly. With these techniques, we aim to reduce to the model size to less than 10 million parameters.

Acknowledgments

The authors acknowledge the Center for Advanced Research Computing (CARC) at the University of Southern California for providing computing resources that have contributed to the research results reported within this publication. URL: https://carc.usc.edu.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *International Conference on Machine Learning*, PMLR, 2017, 214–23.

[2] Z. Azizi, X. Lei, and C.-C. J. Kuo, "Noise-aware Texture-preserving Low-light Enhancement," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, 443–6.

[3] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary Equilibrium Generative Adversarial Networks," *arXiv preprint arXiv:1703.10717*, 2017.

[4] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, "Optimizing the Latent Space of Generative Networks," *arXiv preprint arXiv:1707.05776*, 2017.

[5] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 1986, 679–98.

[6] H. Chang, D. Yeung, and Y. Xiong, "Super-resolution through Neighbor Embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1, IEEE, 2004, I–I.

[7] H. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "Defakehop: A Light-weight High-performance Deepfake Detector," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

[8] H. Chen, K. Zhang, S. Hu, S. You, and C.-C. J. Kuo, "Geo-DefakeHop: High-Performance Geographic Fake Image Detection," *arXiv preprint arXiv:2110.09795*, 2021.

[9] H.-S. Chen, S. Hu, S. You, and C.-C. J. Kuo, "DefakeHop++: An Enhanced Lightweight Deepfake Detector," *arXiv preprint arXiv:2205.00211*, 2022.

[10] Y. Chen and C.-C. J. Kuo, "Pixelhop: A Successive Subspace Learning (ssl) Method for Object Recognition," *Journal of Visual Communication and Image Representation*, 70, 2020, 102749.

[11] Y. Chen, M. Rouhsedaghat, S. You, R. Rao, and C.-C. J. Kuo, "Pixelhop++: A Small Successive-subspace-learning-based (SSL-based) Model for Image Classification," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 3294–8.

[12] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang, "Fsrnet: End-to-end Learning Face Super-resolution with Facial Priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 2492–501.

[13] P. Dhariwal and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis," *Advances in Neural Information Processing Systems*, 34, 2021.

[14] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear Independent Components Estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[15] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density Estimation Using Real NVP," *arXiv preprint arXiv:1605.08803*, 2016.

[16] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a Deep convolutional Network for Image Super-resolution," in *European Conference on Computer Vision*, Springer, 2014, 184–99.

[17] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based Super-resolution," *IEEE Computer Graphics and Applications*, 22(2), 2002, 56–65.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, 27, 2014.

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," *Advances in Neural Information Processing Systems*, 30, 2017.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium," *Advances in Neural Information Processing Systems*, 30, 2017.

[21] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded Diffusion Models for High Fidelity Image Generation," *Journal of Machine Learning Research*, 23(47), 2022, 1–33.

[22] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," *arXiv preprint arxiv:2006.11239*, 2020.

[23] Y. Hoshen, K. Li, and J. Malik, "Non-adversarial Image Synthesis with Generative Latent Nearest Neighbors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 5811–9.

[24] C. Huang, Z. Wang, and C.-C. J. Kuo, "Visible-light and Near-infrared Face Recognition at a Distance," *Journal of Visual Communication and Image Representation*, 41, 2016, 140–53.

[25] J. Johnson, M. Douze, and H. Jégou, "Billion-scale Similarity Search with GPUS," *IEEE Transactions on Big Data*, 7(3), 2019, 535–47.

[26] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "GPCO: An Unsupervised Green Point Cloud Odometry Method," *arXiv preprint arXiv:2112.04054*, 2021.

[27] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "R-pointhop: A Green, Accurate and Unsupervised Point Cloud Registration Method," *arXiv preprint arXiv:2103.08129*, 2021.

[28] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "Unsupervised Point Cloud Registration via Salient Points Analysis (SPA)," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, 5–8.

[29] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," *arXiv preprint arXiv:1710.10196*, 2017.

[30] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training Generative Adversarial Networks with Limited Data," *Advances in Neural Information Processing Systems*, 33, 2020, 12104–14.

[31] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free Generative Adversarial Networks," *Advances in Neural Information Processing Systems*, 34, 2021, 852–63.

[32] T. Karras, S. Laine, and T. Aila, "A Style-based Generator Architecture for Generative Adversarial Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 4401–10.

[33] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of Stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 8110–9.

[34] D. P. Kingma and M. Welling, "Auto-encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[35] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On Convergence and Stability of GANs," *arXiv preprint arXiv:1705.07215*, 2017.

[36] M. Kowalski, S. J. Garbin, V. Estellers, T. Baltrušaitis, M. Johnson, and J. Shotton, "Config: Controllable Neural Face Image Generation," in *European Conference on Computer Vision*, Springer, 2020, 299–315.

[37] C.-C. J. Kuo, "The CNN as a Guided Multilayer RECOS Transform [lecture notes]," *IEEE Signal Processing Magazine*, 34(3), 2017, 81–9.

[38] C.-C. J. Kuo, "Understanding Convolutional Neural Networks with a Mathematical Model," *Journal of Visual Communication and Image Representation*, 41, 2016, 406–13.

[39] C.-C. J. Kuo and Y. Chen, "On Data-driven Saak Transform," *Journal of Visual Communication and Image Representation*, 50, 2018, 237–46.

[40] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable Convolutional Neural Networks via Feedforward Design," *Journal of Visual Communication and Image Representation*, 60, 2019, 346–59.

[41] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic Single Image Super-resolution Using a Generative Adversarial Network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 4681–90.

[42] X. Lei, W. Wang, and C.-C. J. Kuo, "GENHOP: an Image Generation Method Based on Successive Subspace Learning," *IEEE International Symposium on Circuits & Systems (ISCAS)*, 2022.

[43] X. Lei, G. Zhao, and C.-C. J. Kuo, "NITES: A Non-parametric Interpretable Texture Synthesis Method," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2020, 1698–706.

[44] X. Lei, G. Zhao, K. Zhang, and C.-C. J. Kuo, "TGHop: An Explainable, Efficient, and Lightweight Method for Texture Generation," *APSIPA Transactions on Signal and Information Processing*, 10, 2021.

[45] K. Li and J. Malik, "Implicit Maximum Likelihood Estimation," *arXiv preprint arXiv:1809.09087*, 2018.

[46] B. Liu, Y. Zhu, K. Song, and A. Elgammal, "Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis," in *International Conference on Learning Representations*, 2020.

[47] X. Liu, F. Xing, C. Yang, J. Kuo, S. Babu, G. Fakhri, T. Jenkins, and J. Woo, "Voxelhop: Successive Subspace Learning for ALS Disease Classification using Structural MRI," *IEEE Journal of Biomedical and Health Informatics*, 2021.

[48] Y. Lu, Y.-W. Tai, and C.-K. Tang, "Attribute-guided Face Generation Using Conditional CycleGAN," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, 282–97.

[49] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs Created Equal? A Large-scale Study," *Advances in Neural Information Processing Systems*, 31, 2018.

[50] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, and S. Paul Smolley, "Least Squares Generative Adversarial Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2794–802.

[51] M. Monajatipoor, M. Rouhsedaghat, L. H. Li, A. Chien, C. Kuo, F. Scalzo, and K. Chang, "BERTHop: An Effective Vision-and-Language Model for Chest X-ray Disease Diagnosis," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, IEEE, 2021, 3327–36.

[52] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image Transformer," in *International Conference on Machine Learning*, PMLR, 2018, 4055–64.

[53] S. Qian, K.-Y. Lin, W. Wu, Y. Liu, Q. Wang, F. Shen, C. Qian, and R. He, "Make a Face: Towards Arbitrary High Fidelity Face Manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 10033–42.

[54] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating Diverse High-fidelity Images with vq-vae-2," *Advances in neural information processing systems*, 32, 2019.

[55] D. A. Reynolds, "Gaussian Mixture Models," *Encyclopedia of Biometrics*, 741(659-663), 2009.

[56] M. Rouhsedaghat, M. Monajatipoor, Z. Azizi, and C.-C. J. Kuo, "Successive Subspace Learning: An Overview," *arXiv preprint arXiv:2103.00121*, 2021.

[57] M. Rouhsedaghat, Y. Wang, X. Ge, S. Hu, S. You, and C.-C. J. Kuo, "Facehop: A Light-weight Low-resolution Face Gender Classification Method," in *International Conference on Pattern Recognition*, Springer, 2021, 169–83.

[58] M. Rouhsedaghat, Y. Wang, S. Hu, S. You, and C.-C. J. Kuo, "Low-resolution Face Recognition in Resource-constrained Environments," *Pattern Recognition Letters*, 149, 2021, 193–9.

[59] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *science*, 290(5500), 2000, 2323–6.

[60] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image Super-resolution via Iterative Refinement," *arXiv preprint arXiv: 2104.07636*, 2021.

[61] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, *et al.*, "Conditional Image Generation with Pixelcnn Decoders," *Advances in Neural Information Processing Systems*, 29, 2016.

[62] A. Van Den Oord, O. Vinyals, *et al.*, "Neural Discrete Representation Learning," *Advances in Neural Information Processing Systems*, 30, 2017.

[63] J. J. Yu, K. G. Derpanis, and M. A. Brubaker, "Wavelet Flow: Fast Training of High Resolution Normalizing Flows," *Advances in Neural Information Processing Systems*, 33, 2020, 6184–96.

[64] K. Zhang, B. Wang, W. Wang, F. Sohrab, M. Gabbouj, and C.-C. J. Kuo, "Anomalyhop: An SSL-based Image Anomaly Localization Method," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2021, 1–5.

[65] M. Zhang, P. Kadam, S. Liu, and C.-C. J. Kuo, "GSIP: Green Semantic Segmentation of Large-Scale Indoor Point Clouds," *arXiv preprint arXiv:2109.11835*, 2021.

[66] M. Zhang, P. Kadam, S. Liu, and C.-C. J. Kuo, "Unsupervised Feedforward Feature (UFF) Learning for Point Cloud Classification and Segmentation," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, 144–7.

[67] M. Zhang, Y. Wang, P. Kadam, S. Liu, and C.-C. J. Kuo, "Pointhop++: A Lightweight Learning Model on Point Sets for 3d Classification," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 3319–23.

[68] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "Pointhop: An Explainable Machine Learning Method for Point Cloud Classification," *IEEE Transactions on Multimedia*, 22(7), 2020, 1744–55.

[69] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-consistent Adversarial Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2223–32.

[70] Y. Zhu, X. Wang, H. Chen, R. Salloum, and C.-C. J. Kuo, "A-PixelHop: A Green, Robust and Explainable Fake-Image Detector," *arXiv preprint arXiv:2111.04012*, 2021.